**REST API Design for a Blog Application**

**Base URL**

<https://api.example.com/v1>

**Authentication Scheme**

* JWT-based authentication using access and refresh tokens
* Access token in Authorization header: Authorization: Bearer <access\_token>
* Refresh token in cookies (HttpOnly, Secure)

**1. Authentication Endpoints**

**Register a User**

* **Endpoint**: POST /auth/register
* **Request Body**:

{

"username": "sanjeev",

"email": "sanjeev@gmail.com",

"password": "password123",

"confirmPassword": "password123"

}

* **Response**:

{

"message": "User created successfully.",

"status": 201,

"data": {

"id": 101,

"username": "sanjeev",

"email": "sanjeev@gmail.com"

}

}

**Login User**

* **Endpoint**: POST /auth/login
* **Request Body**:

{

"email": "sanjeev@gmail.com",

"password": "password123"

}

* **Response**:

{

"message": "Login successful.",

"status": 200,

"data": {

"accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",

"refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

}

}

**Logout User**

* **Endpoint**: POST /auth/logout
* **Headers**: Authorization: Bearer <access\_token>
* **Response**:

{

"message": "Logged out successfully.",

"status": 200

}

**Refresh Access Token**

* **Endpoint**: POST /auth/refresh
* **Headers**: Authorization: Bearer <refresh\_token>
* **Response**:

{

"message": "Token refreshed successfully.",

"status": 200,

"data": {

"accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

}

}

**2. Password Management**

**Change Password**

* **Endpoint**: POST /auth/change-password
* **Headers**: Authorization: Bearer <access\_token>
* **Request Body**:

{

"currentPassword": "oldpassword123",

"newPassword": "newpassword456",

"confirmNewPassword": "newpassword456"

}

* **Response**:

{

"message": "Password changed successfully.",

"status": 200

}

**Forgot Password**

* **Endpoint**: POST /auth/forgot-password
* **Request Body**:

{

"email": "sanjeev@gmail.com"

}

* **Response**:

{

"message": "Password reset email sent.",

"status": 200

}

**Reset Password**

* **Endpoint**: POST /auth/reset-password
* **Request Body**:

{

"token": "reset\_token\_from\_email",

"newPassword": "newpassword456",

"confirmNewPassword": "newpassword456"

}

* **Response**:

{

"message": "Password reset successfully.",

"status": 200

}

**3. User Endpoints**

**Get All Users (Admin only)**

* **Endpoint**: GET /users
* **Headers**: Authorization: Bearer <access\_token>
* **Query Parameters**:
  + page (optional): Page number for pagination
  + limit (optional): Number of users per page
* **Response**:

{

"message": "Users retrieved successfully.",

"status": 200,

"data": {

"users": [

{

"id": 101,

"username": "sanjeev",

"email": "sanjeev@gmail.com",

"createdAt": "2023-01-01T00:00:00.000Z"

}

],

"pagination": {

"page": 1,

"limit": 10,

"total": 1,

"pages": 1

}

}

}

**Delete User Account**

* **Endpoint**: DELETE /users/:id
* **Headers**: Authorization: Bearer <access\_token>
* **Response**:

{

"message": "User deleted successfully.",

"status": 200

}

**4. Profile Endpoints**

**Get Current User Profile**

* **Endpoint**: GET /me/profile
* **Headers**: Authorization: Bearer <access\_token>
* **Response**:

{

"message": "Profile retrieved successfully.",

"status": 200,

"data": {

"id": 101,

"username": "sanjeev",

"email": "sanjeev@gmail.com",

"bio": "Software developer",

"avatar": "https://example.com/avatar.jpg",

"createdAt": "2023-01-01T00:00:00.000Z",

"updatedAt": "2023-01-01T00:00:00.000Z"

}

}

**Update Current User Profile**

* **Endpoint**: PATCH /me/profile
* **Headers**: Authorization: Bearer <access\_token>
* **Request Body**:

{

"bio": "Updated bio",

"avatar": "https://example.com/new-avatar.jpg"

}

* **Response**:

{

"message": "Profile updated successfully.",

"status": 200,

"data": {

"id": 101,

"username": "sanjeev",

"email": "sanjeev@gmail.com",

"bio": "Updated bio",

"avatar": "https://example.com/new-avatar.jpg",

"createdAt": "2023-01-01T00:00:00.000Z",

"updatedAt": "2023-01-15T10:30:00.000Z"

}

}

**Get Public User Profile**

* **Endpoint**: GET /users/:id/profile
* **Response**:

{

"message": "Profile retrieved successfully.",

"status": 200,

"data": {

"id": 101,

"username": "sanjeev",

"bio": "Software developer",

"avatar": "https://example.com/avatar.jpg",

"createdAt": "2023-01-01T00:00:00.000Z"

}

}

**5. Posts Endpoints**

**List Posts**

* **Endpoint**: GET /posts
* **Query Parameters**:
  + page (optional): Page number
  + limit (optional): Number of posts per page
  + authorId (optional): Filter by author ID
  + tag (optional): Filter by tag
  + search (optional): Search in title and content
  + sortBy (optional): Field to sort by (e.g., createdAt, title)
  + order (optional): Sort order (asc or desc)
* **Response**:

{

"message": "Posts retrieved successfully.",

"status": 200,

"data": {

"posts": [

{

"id": 201,

"title": "First Post",

"excerpt": "This is the first post...",

"author": {

"id": 101,

"username": "sanjeev"

},

"tags": ["tech", "programming"],

"createdAt": "2023-01-01T00:00:00.000Z",

"updatedAt": "2023-01-01T00:00:00.000Z",

"likeCount": 5,

"commentCount": 3

}

],

"pagination": {

"page": 1,

"limit": 10,

"total": 1,

"pages": 1

}

}

}

**Get a Post by ID**

* **Endpoint**: GET /posts/:id
* **Response**:

{

"message": "Post retrieved successfully.",

"status": 200,

"data": {

"id": 201,

"title": "First Post",

"content": "This is the full content of the first post...",

"author": {

"id": 101,

"username": "sanjeev"

},

"tags": ["tech", "programming"],

"createdAt": "2023-01-01T00:00:00.000Z",

"updatedAt": "2023-01-01T00:00:00.000Z",

"likeCount": 5,

"commentCount": 3

}

}

**Create a Post**

* **Endpoint**: POST /posts
* **Headers**: Authorization: Bearer <access\_token>
* **Request Body**:

{

"title": "New Post",

"content": "This is the content of the new post...",

"tags": ["tech", "new"]

}

* **Response**:

{

"message": "Post created successfully.",

"status": 201,

"data": {

"id": 202,

"title": "New Post",

"content": "This is the content of the new post...",

"author": {

"id": 101,

"username": "sanjeev"

},

"tags": ["tech", "new"],

"createdAt": "2023-01-15T10:30:00.000Z",

"updatedAt": "2023-01-15T10:30:00.000Z",

"likeCount": 0,

"commentCount": 0

}

}

**Update a Post**

* **Endpoint**: PUT /posts/:id
* **Headers**: Authorization: Bearer <access\_token>
* **Request Body**:

{

"title": "Updated Post",

"content": "Updated content...",

"tags": ["tech", "updated"]

}

* **Response**:

{

"message": "Post updated successfully.",

"status": 200,

"data": {

"id": 202,

"title": "Updated Post",

"content": "Updated content...",

"author": {

"id": 101,

"username": "sanjeev"

},

"tags": ["tech", "updated"],

"createdAt": "2023-01-15T10:30:00.000Z",

"updatedAt": "2023-01-15T11:30:00.000Z",

"likeCount": 0,

"commentCount": 0

}

}

**Delete a Post**

* **Endpoint**: DELETE /posts/:id
* **Headers**: Authorization: Bearer <access\_token>
* **Response**:

{

"message": "Post deleted successfully.",

"status": 200

}

**Get Logged-in User's Posts**

* **Endpoint**: GET /me/posts
* **Headers**: Authorization: Bearer <access\_token>
* **Query Parameters**:
  + page (optional): Page number
  + limit (optional): Number of posts per page
* **Response**: Same as List Posts response

**Get Posts from a Specific User**

* **Endpoint**: GET /users/:id/posts
* **Query Parameters**:
  + page (optional): Page number
  + limit (optional): Number of posts per page
* **Response**: Same as List Posts response

**6. Comments Endpoints**

**Add Comment to a Post**

* **Endpoint**: POST /posts/:id/comments
* **Headers**: Authorization: Bearer <access\_token>
* **Request Body**:

{

"content": "This is a comment on the post."

}

* **Response**:

{

"message": "Comment added successfully.",

"status": 201,

"data": {

"id": 301,

"content": "This is a comment on the post.",

"author": {

"id": 101,

"username": "sanjeev"

},

"postId": 201,

"createdAt": "2023-01-15T12:30:00.000Z",

"updatedAt": "2023-01-15T12:30:00.000Z"

}

}

**List Comments for a Post**

* **Endpoint**: GET /posts/:id/comments
* **Query Parameters**:
  + page (optional): Page number
  + limit (optional): Number of comments per page
* **Response**:

{

"message": "Comments retrieved successfully.",

"status": 200,

"data": {

"comments": [

{

"id": 301,

"content": "This is a comment on the post.",

"author": {

"id": 101,

"username": "sanjeev"

},

"postId": 201,

"createdAt": "2023-01-15T12:30:00.000Z",

"updatedAt": "2023-01-15T12:30:00.000Z"

}

],

"pagination": {

"page": 1,

"limit": 10,

"total": 1,

"pages": 1

}

}

}

**Update a Comment**

* **Endpoint**: PUT /comments/:id
* **Headers**: Authorization: Bearer <access\_token>
* **Request Body**:

{

"content": "Updated comment content."

}

* **Response**:

{

"message": "Comment updated successfully.",

"status": 200,

"data": {

"id": 301,

"content": "Updated comment content.",

"author": {

"id": 101,

"username": "sanjeev"

},

"postId": 201,

"createdAt": "2023-01-15T12:30:00.000Z",

"updatedAt": "2023-01-15T13:30:00.000Z"

}

}

**Delete a Comment**

* **Endpoint**: DELETE /comments/:id
* **Headers**: Authorization: Bearer <access\_token>
* **Response**:

json

{

"message": "Comment deleted successfully.",

"status": 200

}

**Get All Comments by Current User**

* **Endpoint**: GET /me/comments
* **Headers**: Authorization: Bearer <access\_token>
* **Query Parameters**:
  + page (optional): Page number
  + limit (optional): Number of comments per page
* **Response**: Same as List Comments response

**7. Likes Endpoints**

**Like a Post**

* **Endpoint**: POST /posts/:id/likes
* **Headers**: Authorization: Bearer <access\_token>
* **Response**:

{

"message": "Post liked successfully.",

"status": 200

}

**Unlike a Post**

* **Endpoint**: DELETE /posts/:id/likes
* **Headers**: Authorization: Bearer <access\_token>
* **Response**:

{

"message": "Post unliked successfully.",

"status": 200

}

**Get Total Likes for a Post**

* **Endpoint**: GET /posts/:id/likes
* **Response**:

{

"message": "Likes retrieved successfully.",

"status": 200,

"data": {

"postId": 201,

"likeCount": 5

}

}

**Check if Current User Liked a Post**

* **Endpoint**: GET /posts/:id/likes/me
* **Headers**: Authorization: Bearer <access\_token>
* **Response**:

{

"message": "Like status retrieved successfully.",

"status": 200,

"data": {

"postId": 201,

"hasLiked": true

}

}

**Get Posts Liked by Current User**

* **Endpoint**: GET /me/likes
* **Headers**: Authorization: Bearer <access\_token>
* **Query Parameters**:
  + page (optional): Page number
  + limit (optional): Number of posts per page
* **Response**: Same as List Posts response

**Error Responses**

**Authentication Error**

{

"message": "Authentication failed.",

"status": 401,

"error": "Invalid token"

}

**Authorization Error**

{

"message": "Access denied.",

"status": 403,

"error": "Insufficient permissions"

}

**Not Found Error**

{

"message": "Resource not found.",

"status": 404,

"error": "Post not found"

}

**Validation Error**

{

"message": "Validation failed.",

"status": 422,

"error": "Invalid email format",

"details": [

{

"field": "email",

"message": "Invalid email format"

}

]

}